Explanation of how User Subscription/Notification works:

The Subscription/Notification system consists of:

* Database Subscription/Notification table.
* JS component to alerts users on subsequent new notification.
* A pluggable backend system

The system is using SQLAlchemy to access and manage data in the SQLite database. SQLAlchemy is adapted into pythonic domain language.

In the SQLALchemy python code file, Session is established to connect with the database and represents for all objects you’ve loaded or associated with it during the lifespan. The Session includes add() method, commit() method, delete() method and close() method. add() is use to place instances in the session. commit() is use to write changes to the database. delete() is use to place an instance into the Session’s list of objects to be marked as deleted.

Controllers consists of the python code of the Subscribe and Unsubscribe. Those forms are using Flask as a microframework, which is using jinja as a template engine.

The component, sockethandler.js, enables all communication to go through one socket. This is dependent on the main jinja template which depends on the Flask-SocketIO.

In the Flask code, the url\_for() function will build a Url for a specific function. The render\_template() function will render a HTML file which base on the jinja and html code. The get\_flashed\_message() passes a message to the next request, which generally is a template.

The templates folder consists of main.html and PostView.html which are based on jinja and html code. The CSS of the html files will be using the bootstrap library.

Controllers file, routes.py, templates files are all included in app file.